

Date August 31, 2010

To Examiner Hang Pan

Of U.S. Patent and Trademark Office

Fax 571-270-8667

From Mark J. DeBoy

Subject Topics For Examiner Interview

Our Ref Q86474 Your Ref 10/525,382

Pages 4  
(including cover sheet)

---

Please call attention to problems with this transmission by return fax or telephone. Thank you.

THE INFORMATION CONTAINED IN THIS COMMUNICATION IS CONFIDENTIAL, MAY BE ATTORNEY-CLIENT PRIVILEGED, AND IS INTENDED ONLY FOR THE USE OF THE ADDRESSEE. UNAUTHORIZED USE, DISCLOSURE OR COPYING IS STRICTLY PROHIBITED AND MAY BE UNLAWFUL. IF YOU HAVE RECEIVED THIS COMMUNICATION IN ERROR, PLEASE IMMEDIATELY NOTIFY US.

FOR DISCUSSION PURPOSE ONLY. NOT FOR ENTRY IN APPLICATION FILE

Dear Examiner Pan,

Please find the attached agenda for our proposed interview. I look forward to hearing from you.

Best regards,  
Mark DeBoy

**FOR DISCUSSION PURPOSES ONLY.  
NOT FOR ENTRY IN APPLICATION FILE.**

**ANNEX A Arguments on difference between Kohl and present invention**

This is a final effort to highlight the differences between the mechanism taught by Kohl and the own mechanism.

All database tables consist of columns and rows. Here is an example of a table containing data about homes.

Id	Name	Owner	Stars	Active	Internal notes
1	Blue eagle	37	3	yes	Will be repainted this summer
2	Sunset	68	4	no	Old-fashioned furniture
3	Indian summer	68	4	yes	

Now consider authorizations. There are 2 types of authorizations:

- **column authorizations**  
A user may be authorized to read certain columns only. For example, a user is allowed to see all columns, except the **Internal notes** column.
- **row authorizations**  
A user may be authorized to read certain rows. For example a user is allowed to read row 1, but not row 2 and 3.

(Naturally, we can also consider the combination of these two types, but that is not necessary for this plea.)

Let's focus on the differences between these two types of authorizations.

**Column authorizations** can be defined for certain user groups. An automated system can apply these authorizations without looking at the data in the table. Data about the individual user is also not needed. The automated system always applies the authorizations without knowing the actual data.

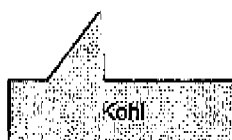
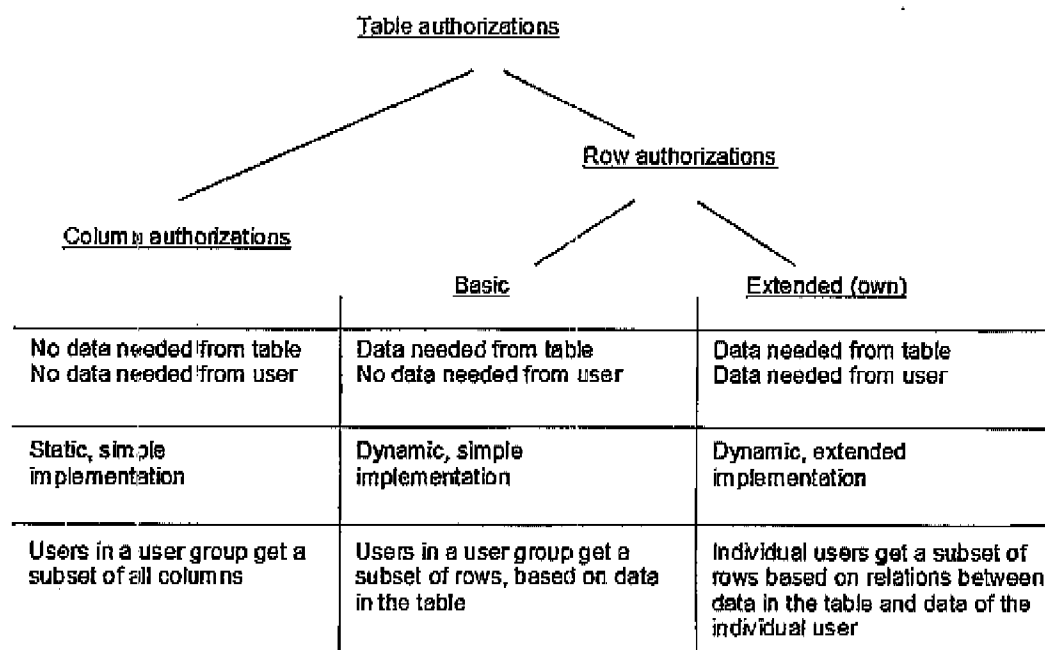
**Row authorizations** can be basic and complex.

- **Basic row authorizations**  
These authorizations can also be defined for certain user groups and are based on a filter, like "active = yes". This means that certain users (e.g. visitors of a website) are only allowed to see houses which are active. The automated system needs the data in the table, but does not need data about the individual user.
- **Extended row authorizations**  
These authorizations use an actual relationship between the data in the table and

**FOR DISCUSSION PURPOSES ONLY.  
NOT FOR ENTRY IN APPLICATION FILE.**

the data of the user. In this example, users are only allowed to see data of their own houses. Suppose a user has key value 37. A look at the table shows that this person is the owner of home 'Blue eagle'. This means that this user is only allowed to see the data of row 1. Since the user does not own the other houses, this user is not allowed to read the data in the other rows. In this case we talk about **own** permissions.

To summarize, **column authorizations** never have a relation to actual data and are defined for groups of users, not for individual users. Once set, they are static and they remain static. **Row authorizations** are more dynamic. **Basic row authorizations** depend on actual data in the table, while **extended row authorizations** depend on actual data in the table and actual data of the particular user. The working principle of the latter is very different: the automated system needs actual user data, and has to compare this data with the data in the table in order to apply the authorizations correctly.



**FOR DISCUSSION PURPOSES ONLY.  
NOT FOR ENTRY IN APPLICATION FILE.**

We argue that Kohl only teaches **column** **authorizations**, and does not teach row authorizations. Let's have a closer look at what Kohl teaches:

- present different salespeople with different sets of the **columns** (Column 7, line 53-54).
- Salespeople in an LA office may be presented with an "Orders" application where the "Ship from" **column** is not visible (Column 7, line 55-57)
- Salespeople in the field may, however, be presented with an "Orders" application where the "Ship from" **column** is available (Column 7, line 57-59)

In all examples Kohl only mentions columns. Kohl never mentions rows. Talking about rows is **essential** to explain the **own** mechanism. Without talking about rows, Kohl cannot refer to or explain the mechanism behind the **own** permission (not even in the broadest interpretation).

Furthermore, Kohl never mentions individual users and authorizations based on their personal data. Kohl only talks about groups of users. When only groups of users are mentioned, the mechanism can never relate to the **own** method.